

2024- 2025
PROJET DE FIN D'ÉTUDES

DIPLÔME NATIONAL D'INGÉNIEUR

SPÉCIALITÉ : INFORMATIQUE

ChatBot As A Service

Réalisé par: CHADINE BEN AISSA

Encadré par:

Encadrant ESPRIT: FAROUK JELIDI

Encadrant Entreprise: AMINE GONGI

INSOMEA
COMPUTER SOLUTIONS

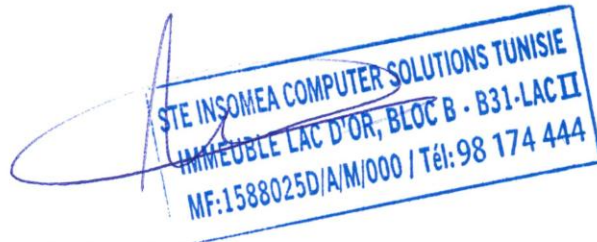
Je valide le dépôt du rapport PFE relatif à l'étudiant nommé ci-dessous / I
validate the submission of the student's report:

- Nom & Prénom /Name & Surname :*Chadine Ben Aissa*.....

Encadrant Entreprise/ Business site Supervisor

- Nom & Prénom /Name & Surname :*Amine Gargi*.....

Cachet & Signature / Stamp & Signature



Encadrant Académique/Academic Supervisor

- Nom & Prénom /Name & Surname :*Faruk JELIDI*.....

Signature / Signature

A green ink signature, appearing to be 'F. J.', is written below the signature line.

Ce formulaire doit être rempli, signé et scanné/This form must be completed, signed and scanned.

Ce formulaire doit être introduit après la page de garde/ This form must be inserted after the cover page.

Dédicace

À mon père, **Abdelghani Ben Aissa**, pour son soutien indéfectible, ses sacrifices et sa sagesse qui m'a continuellement accompagnée.

À ma mère, **Soumaya Chaieb**, pour son amour sa patience et son soutien inébranlable

À mes frères et ma sœur, **Ahmed , Mohamed et Syrine Ben Aissa** pour leur présence précieuse, leurs encouragements et leur confiance en moi.

À mes proches et amis, pour leur accompagnement sans faille et leurs encouragements et leur présence précieuse dans les moments d'échec comme dans les réussites.

À l'équipe pédagogique et administrative d'ESPRIT, pour m'avoir offert un milieu propice à l'apprentissage, ouvert sur l'innovation et l'excellence.

À **M. Farouk Jelidi**, mon encadrant académique, pour sa bienveillance, sa guidance experte et son suivi constant à chaque phase du projet

À **M. Amine Gongi**, mon encadrant professionnel, pour sa confiance, sa disponibilité et l'apport professionnel dont j'ai bénéficié grâce à lui

Cette modeste réalisation vous est dédiée, avec toute ma reconnaissance et mon respect

Remerciements

Je suis profondément reconnaissante envers toutes les personnes qui m'ont accompagnée et soutenue tout au long de ce projet de fin d'études. Je remercie tout d'abord mon encadrant académique, **Mr. Farouk Jelidi**, pour ses conseils, sa présence et ses encouragements constants. Son encadrement précis et soutenu a été un facteur très intéressant pour la réussite de ce travail.

Je remercie également mon maître de stage, **Mr. Amine Gongi**, pour m'avoir accueillie au sein de **Insomea**, pour sa confiance, son encadrement et la richesse des échanges professionnels et humains tout au long du projet .

Un grand merci à toute l'équipe technique avec laquelle j'ai eu le plaisir de travailler, pour leur bienveillance, leur esprit d'équipe et leur disponibilité.

Enfin, Je remercie chaleureusement ma famille et mes proches pour l'appui moral qu'ils m'ont apporté tout au long de cette formation.



Table Des Matières

Dédicace	2
Remerciements	6
Introduction Générale	13
1. Chapitre 1 : Présentation Générale de Projet	15
1.1 Présentation de l'organisme d'accueil	15
1.1.1 Introduction	15
1.1.2 Domaine d'activité	15
1.1.3 Organisme Interne	16
1.2 Contexte Générale du projet	17
1.2.1 Introduction	17
1.2.2 Etude de l'existant	17
1.2.3 Critique de l'existant	18
1.3 Problématique	19
1.4 Solution Proposée	19
1.5 Choix de la méthodologie	20
1.5.1 La Méthode SCRUM	20
1.5.2 Les Composants de la Méthode AGILE	20
1.6 Conclusion	21
2. Chapitre 2 : Sprint 0 : Analyse et spécification des besoins	23
2.1 Introduction	23
2.2 Etude de contexte	23
2.2.1 Identification des acteurs	23
2.2.2 Les besoins fonctionnelles	24
2.2.3 Les besoins non fonctionnelles	24
2.3 Product Backlog	25
2.4 Diagramme de cas d'utilisation global	28
2.5 Diagramme de Classe	29
2.6 Architecture Globale	30
2.6.1 Architecture Logique	30
2.6.2 Architecture Physique	31

2.7 Environnement de travail	32
2.7.1 Outils et Technologies utilisées	32
2.8 Planification des Sprints	34
2.9 Conclusion	35
3. Chapitre 3 : Sprint 1 : Authentification et Gestion des utilisateurs	37
3.1 Introduction	37
3.2 Analyse et spécification des besoins	37
3.2.1 Sprint Backlog	37
3.3 Diagramme de cas d'utilisation Sprint 1	38
3.4 Diagramme d'activité Sprint 1	39
3.5 Diagramme de séquence système Sprint 1	40
3.6 Conception	41
3.7 Réalisations	42
3.8 Conclusion	45
4. Chapitre 4 : Sprint 2 : Gestion des Chatbots et des Templates	47
4.1 Introduction	47
4.2 Analyse et spécification des besoins	47
4.2.1 Sprint Backlog	47
4.3 Diagramme de cas d'utilisation Sprint 2	48
4.4 Diagramme de séquence Sprint 2	48
4.5 Réalisations	49
4.6 Conclusion	49
5. Chapitre 5 : Sprint 3 : Formulaire Dynamique et Génération des fichiers Terraform	51
5.1 Introduction	51
5.2 Analyse et spécification des besoins	51
5.2.1 Les Champs Du Formulaire	51
5.2.2 Les Besoins Utilisateurs	52
5.2.3 Les Besoins Administrateur	52
5.2.4 Sprint Backlog	53
5.3 Diagramme de cas d'utilisation et de Séquence	54
5.3.1 Diagramme de cas d'utilisation Sprint 3	54
5.3.2 Diagramme de séquence Sprint 3	55
5.4 Génération des Fichiers Terraform	56
5.4.1 Objectif	56
5.4.2 Les Composants Générés	56

5.4.3 La Logique de Génération	56
5.4.4 Les Points Importants	56
5.5 Réalisations	57
5.6 Conclusion	59
 6. Chapitre 6 : Sprint 4 : Déploiement Dynamique sur Azure via Terraform	 61
6.1 Introduction	61
6.2 Analyse et spécification des besoins	61
6.2.1 Sprint Backlog	61
6.3 Diagramme de cas d'utilisation et de Séquence	63
6.3.1 Diagramme de cas d'utilisation Sprint 4	63
6.3.2 Diagramme de séquence Sprint 4	64
6.4 Formulaire de Déploiement	64
6.4.1 Objectif	64
6.4.2 Les Champs du Formulaire	64
6.4.3 La Logique de Validation	64
6.4.4 Les Points Importants	65
6.5 Réalisations	66
6.6 Conclusion	66
 7. Conclusion Générale	 67
 8. Références Bibliographiques	 68
 9. Annexes	 69

Table des Figures

Figure 1.1 : Organigramme de l'entreprise INSOMEA	15
Figure 1.2 : Organigramme de l'organisation Interne d'INSOMEA	16
Figure 1.3 : Microsoft Bot Framework	17
Figure 1.4: Diagflow	18
Figure 1.5 : Comparaison des Plateformes	18
Figure 1.6 : SCRUM Process	20
Figure 2.1 : Identifications des Acteurs	23
Figure 2.2 : Diagramme de cas d'utilisation global	28
Figure 2.3 : Diagramme de classe Sprint 0	29
Figure 2.4 : Architecture Logique	30
Figure 2.5: Architecture Physique	31
Figure 3.1: Diagramme de cas d'utilisation Sprint 1.....	38
Figure 3.2: Diagramme d'activité Création d'un compte	39
Figure 3.3: Diagramme séquence système	40
Figure 3.4 : Diagramme séquence objet « s'authentifier »	41
Figure 3.5 : Authentification.....	42
Figure 3.6: Créer un Compte	42
Figure 3.7: Dashboard Admin	43
Figure 3.8 : Dashboard User	43
Figure 3.9: Dashboard Admin : Gestion utilisateur	44

Table des Figures

Figure 3.10: Dashboard Admin : Gestion des codes	44
Figure 4.1: Diagramme de cas d'utilisation Sprint 2.....	48
Figure 4.2 : Diagramme séquence objet Sprint 2.....	48
Figure 4.3 : Premier chatbot réalisé avec RAG	49
Figure 5.1 : Diagramme de cas d'utilisation Sprint 3	54
Figure 5.2 : Diagramme de séquence objet Sprint 3	55
Figure 5.3 : Formulaire de Configuration	57
Figure 5.4 : Emplacements Chatbots Générés	57
Figure 5.5 : Template Terraform	58
Figure 6.1 : Diagramme de cas d'utilisation Sprint 4.....	63
Figure 6.2 : Diagramme de séquence Sprint 4	64
Figure 6.3 : Formulaire de Déploiement	66
Figure 6.4: Chatbot en cours de déploiement	66

Table des Tables

Table 2.1 : Product Backlog	25
Table 2.2 : Outils et Technologies Utilisées	32
Table 2.3 : Planification des Sprints	34
Table 2.4 : Sprint Backlog Sprint 1.....	37
Table 2.5 : Procédure d'authentification	38
Table 2.6 : Ajouter Un Utilisateur	39
Table 3.1 : Sprint Backlog Sprint 2	47
Table 4.1: Sprint Backlog Sprint 3	53
Table 5.1: Sprint Backlog Sprint 4	62

Introduction Générale

Dans un monde où la numérisation prend de plus en plus de place, les entreprises, administrations et organisations sont des clients à la recherche de solutions innovantes leur permettant d'automatiser et de personnaliser leur communication avec les utilisateurs. Les chatbots intelligents, capables d'échanger de façon naturelle et multicanal (texte, voix, images, documents...), sont l'une des

Clés de voûte de cette transformation digitale. Leur usage va du support client à la prise de rendez-vous, en passant par l'aide à la décision ou l'assistance virtuelle. Cependant, la réalisation technique de ces agents conversationnels demeure un véritable casse-tête, tant sur le plan technique que sur le plan des compétences. En effet, cela fait appel à des compétences en développement, en intelligence artificielle (IA), en gestion d'infrastructure réseau. Et ce sont des utilisateurs, très majoritairement issus des domaines non techniques, qui se les posent. Une problématique nous semble donc centrale :

Comment des utilisateurs non techniques pourraient-ils concevoir, configurer, déployer des chatbots qui soient intelligents tout en leur assurant une technologie leur permettant de garder la qualité et les possibilités offertes par les progrès technologiques ?

Ce projet de fin d'études désire, tout d'abord, poser ce postulat de base. Le défi soutenu sera matérialisé par le développement d'une plateforme universelle de création et de configuration de chatbots basée sur une architecture de micro services et pensée « cloud », et notamment Azure. Elle aura pour but premier :

- De permettre à des utilisateurs de créer dynamiquement les saisies et les sorties de données en

- Fonction des besoins définis ;

- Emploie des techniques d'intelligence

- artificielle, comme le RAG (Retrieval-Augmented Generation) avec Azure OpenAI, pour fournir des réponses pertinentes

- Permet un déploiement cloud aisé via des outils d'automatisation (Azure DevOps, Terraform, AppServices).

La tentative consiste à comprendre en profondeur les besoins des utilisateurs, à modéliser l'architecture technique, à développer par itérations les briques fonctionnelles, et à procéder à de l'intégration continue sur un environnement de production. C'est un travail autant technique que fonctionnel, puisqu'il s'agit d'implémenter des architectures modernes des choix à une expérience utilisateur simplifiée.

Chapitre 1

1. Présentation du cadre général du projet

1.1 Présentation de l'organisme d'accueil	15
1.1.1 Introduction	15
1.1.2 Domaine d'activité	15
1.1.3 Organisme Interne	16
1.2 Contexte Générale du projet	17
1.2.1 Introduction	17
1.2.2 Etude de l'existant	17
1.2.3 Critique de l'existant	18
1.3 Problématique	19
1.4 Solution Proposée	19
1.5 Choix de la méthodologie	20
1.5.1 La Méthode SCRUM	20
1.5.2 Les Composants de la Méthode AGILE	20
1.6 Conclusion	21

1. Présentation du cadre général du projet

1.1 Présentation de l'organisme d'accueil

1.1.1 Introduction

L'entreprise **INSOMEA Computer Solutions**, fondée en Tunisie, est spécialisée dans la modernisation numérique, le déploiement de solutions Microsoft, le cloud Computing (notamment Microsoft Azure), et la cybersécurité.

Elle accompagne leurs clients dans leur transformation digitale offrant des services allant de l'accompagnement stratégique à l'intégration de solutions innovantes

1.1.2 Domaine d'activité

INSOMEA est un partenaire certifié Microsoft, acteur proposant des solutions cloud pour :

- La transition vers le cloud (Azure)
- La sécurité informatique (SOC, cybersécurité Microsoft Defender)
- La productivité et collaboration (Microsoft 365, Teams, SharePoint)
- La gestion de projet agile et support technique

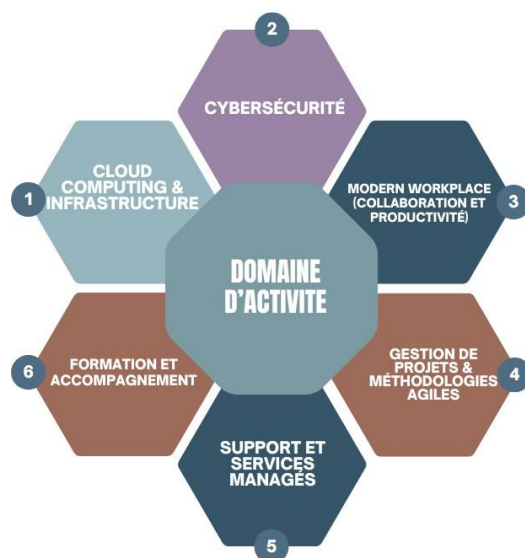


FIGURE 1.1 – Organigramme de l'entreprise INSOMEA

- Workplace

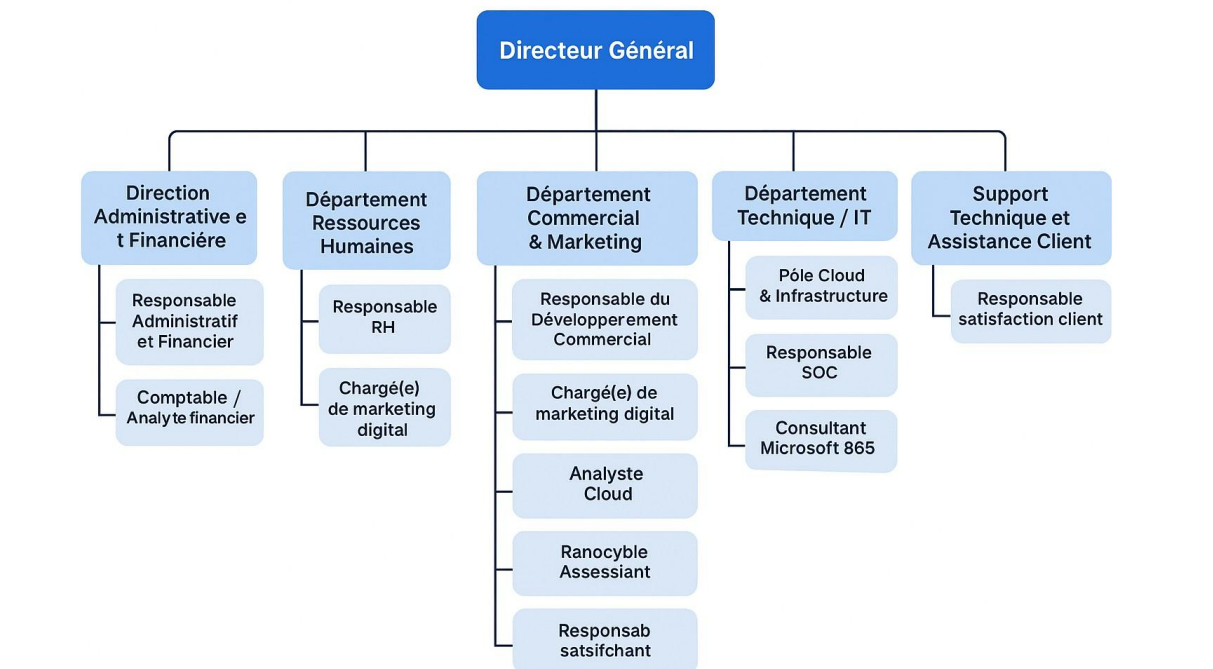


FIGURE 1.2 – Organigramme de l'organisation interne d'INSOMEA

1.2 Contexte Général du projet

1.2.1 Introduction au Sujet :

La mission liée à ce projet est de mettre en place une plateforme pour concevoir et paramétrer des chatbots utilisables dans différents scénarios. Dans les paragraphes suivants, nous allons présenter le contexte du projet, définir les objectifs principaux, et décrire les actions nécessaires pour proposer une solution intuitive, flexible, et accessible sans compétences techniques spécifiques.

1.2.2 Étude de l'existant

À l'heure actuelle, plusieurs plateformes permettent la création et la configuration de chatbots. Cependant, la plupart d'entre elles sont limitées en termes de flexibilité, de personnalisation ou d'accessibilité pour les utilisateurs non techniques. Nous allons à présent étudier les solutions existantes les plus répandues dans ce domaine afin d'identifier leurs avantages et leurs inconvénients.

Microsoft Bot Framework : Le Microsoft Bot Framework est une solution performante et adaptable au création de chatbots complexes intégrant des services avancés comme l'IA ou la reconnaissance vocale. Toutefois, son principal inconvénient se trouve dans sa complexité : sa mise en œuvre requiert des compétences techniques avancées, ce qui le rend peu accessible aux utilisateurs non développeurs.



FIGURE1.3 – Microsoft Bot Framework

Dialogflow (Google) : Dialogflow de Google offre un excellent traitement du langage naturel, facilitant la création d'agents conversationnels performants. Cependant, il dépend fortement de Google Cloud, ce qui peut limiter la flexibilité et entraîner des coûts liés à cette plateforme



FIGURE 1.4 – Dialogflow

1.2.3 Critique de l'existant :

Plateforme	Dépendance Cloud	Personnalisation NLP	Outils visuels workflow	Complexité conception	Payant
Dialogflow (Google)	✓	✗	✓	✗	✓
Bot Framework (Microsoft)	✗	✓	✗	✓	✓

FIGURE1.5 – Comparaison des Plateformes

1.3 Problématique

La volonté d'offrir plus d'autonomie aux utilisateurs et renforcer l'efficacité des opérations des équipes techniques d'**INSOMEA** a rapidement mis en évidence les limites de processus existant

En effet la création de chaque chatbot nécessitait encore trop de temps, interventions humaines et des compétences techniques avancées.

Chaque fois qu'un client demandait un nouveau chatbot les équipes devaient tout recommencer depuis le début : la configuration, le déploiement, l'intégration

Un **travail répétitif, long** souvent **sources d'erreurs**.

Résultat : Délais rallongés charge de travail important, manque d'efficacité dans les livraisons

C'est face à cette réalité que l'idée a émergé et si on pouvait créer une plateforme capable de générer automatiquement des chatbots intelligents sans avoir à tout refaire à chaque fois et sans expertise techniques

De là est née la problématique du projet : **Comment automatiser la création et le déploiement des chatbots intelligents pour réduire le temps de développement, réduire les erreurs humaines et rendre le processus accessible sans expertise technique ?**

1.4 Solution Proposée

Pour éviter les limites des plateformes existantes telles que Dialogflow ou Bot Framework souvent dépendantes d'un cloud unique, peu flexibles ou complexes à personnaliser

Notre projet propose une solution plus ouverte, modulaire et accessible, grâce à une interface conviviale. sans code, les utilisateurs peuvent générer et configurer et déployer des chatbots adaptés à différents besoins sans compétences techniques.

L'architecture microservices simplifie la maintenance et l'évolution du système ,tandis que le déploiement sur Azure via Terraform assure rapidité et fiabilité. La plateforme prend en charge divers types d'entrée (texte, fichiers, images) et intègre une intelligence RAG pour des réponses précises et contextuelles.

Enfin, un dashboard administrateur centralise la gestion, rendant la solution globale, évolutive et parfaitement calibrée pour l'équipe technique

1.5 Choix de la Méthodologie

1.5.1 Méthode SCRUM :

Dans le cadre de notre projet portant sur la création d'une plateforme chez Insomea il est impossible de connaître et figer toutes les spécifications du produit final dès le départ. Cela signifie que la définition des besoins fonctionnels est susceptible d'évoluer et nécessite une démarche participative des acteurs techniques et fonctionnels. Il est donc indispensable de travailler en aveugle technique et fonctionnel. Dans ce cas il faut effectuer un suivi régulier des acteurs par des revues de développement, réajuster de priorités à la réalité des besoins, voire valider la conception. Il convient d'adopter une démarche incrémentale et répétitive dans laquelle chaque membre pourra être force de proposition et où la souplesse face aux besoins de l'entreprise est de mise. C'est pourquoi nous avons choisi de travailler en modes AGILE qui correspond à une réalité optimale pour la conduite de projets de cette nature.

1.5.2 Les Composants de la méthodologie SCRUM :

Trois acteurs clés interviennent dans la répartition des rôles selon la méthodologie Scrum :

— **Product Owner** : plan directeur du produit et responsable de la définition des besoins fonctionnels. Dans notre projet, ce rôle est attribué à l'entreprise Insomea, qui exprime les besoins liés à la plateforme de création de chatbots.

— **Scrum Master** : chargé de simplifier la méthode Scrum, d'organiser les réunions quotidiennes, de lever les obstacles du projet. Dans notre cas, ce rôle est assuré par Mr Amine Gongi encadrant technique chez Insomea.

— **Équipe de développement** : chargé de la conception du projet, du développement et de la mise en œuvre technique du projet. Cette tâche est assurée par les membres de l'équipe projet



FIGURE 1.6 – SCRUM

1.6 Conclusion

Ce chapitre a pour but de donner un aperçu de l'entreprise d'accueil, de clarifier les missions à accomplir, et d'introduire les choix méthodologiques ayant guidé notre démarche. Il forme ainsi un socle stable pour comprendre le contexte global du projet.

L'étude approfondie des besoins fonctionnels et techniques sera présentée dans la section suivante, afin de mieux cerner les attentes et objectifs spécifiques du système à développer.

Chapitre 2

Sprint 0 : Analyse et spécification des besoins

2.1 Introduction	23
2.2 Etude de contexte	23
2.2.1 Identification des acteurs	23
2.2.2 Les besoins fonctionnelles	24
2.2.3 Les besoins non fonctionnelles	24
2.3 Product Backlog	25
2.4 Diagramme de cas d'utilisation global	28
2.5 Diagramme de Classe	29
2.6 Architecture Globale	30
2.6.1 Architecture Logique	30
2.6.2 Architecture Physique	31
2.7 Environnement de travail	32
2.7.1 Outils et Technologies utilisées	32
2.8 Planification des Sprints	34
2.9 Conclusion	35

Sprint 0 : Analyse et Spécification des besoins

2.1 Introduction

Ce chapitre vise à présenter le Sprint 0, une phase fondamentale pour le lancement structuré de notre projet par la méthodologie Scrum . Nous débuterons par la détermination des acteurs majeurs de la solution chatbot, puis nous dresserons la liste des spécifications fonctionnelles et non fonctionnelles spécifiques à notre plateforme. Ces éléments constitueront le socle du backlog produit, en cohérence avec les besoins exprimés par INSOMEA. Par la suite, nous définirons l'architecture technique et logicielle de notre solution, accompagnée d'une étude approfondie de faisabilité. Enfin, ce chapitre se terminera par la planification des sprints, marquant le découpage progressif de notre travail en itérations agiles.

2.2 Étude de contexte

2.2.1 Identification des acteurs :

Identification des acteurs	
ACTEURS	ROLES
ADMINISTRATEUR 	supervise l'ensemble du système. Il gère les templates disponibles, contrôle les activités via le dashboard, et suit les interactions entre utilisateurs et chatbots. Il administre également les comptes et les rôles pour garantir une utilisation fluide, sécurisée et adaptée aux besoins des utilisateurs.
UTILISATEUR 	créer et configurer son propre chatbot via une interface intuitive. Il sélectionne un template, personnalise les paramètres selon ses besoins, et peut tester les interactions pour adapter le comportement du chatbot à son cas d'usage.

FIGURE 2.1 – Identification des acteurs

2.2.2 Les Besoins fonctionnels :

Authentification :

- Authentifier les utilisateurs via des tokens JWT (JSON Web Token).
- Gérer l'inscription et la connexion des utilisateurs.
- Gestion de la partie RBAC

Gestion des utilisateurs :

- Inscription et authentification sécurisée des utilisateurs
- Gestion des rôles (utilisateur, administrateur)
- Gestion des droits d'accès selon les rôles

Gestion des Templates :

- Consultation et sélection de templates : Ajouter, modifier et suppr de templates (Admin)

Gestion d'approvisionnement de chatbots :

- Créer des chatbots
- Configurer les chatbots suit à la création ,
- Déployer l'infrastructure des chatbots directement sur Azure
- Télécharger les fichiers de déploiement Terraform Azure

Dashboard :

2.2.3 Les Besoins non fonctionnels :

Les besoins non fonctionnels que notre application doit prendre en charge sont énumérés dans cette section.

Dans cette section, nous trouvons une liste des exigences non fonctionnelles que notre application doit garantir à l'utilisateur . Les besoins non fonctionnels de l'application sont résumés ci-dessous :

Performance : Botpilot doit gérer jusqu'au 10 milles utilisateurs sans dégradation de performance garantissant la fluidité

Réponse rapide de l'interface (moins de 2 secondes pour les actions principales).



Sécurité : Accès utilisateur sécurisé via OAuth 2.0 / Azure AD.
Chiffrement des données sensibles (en transit et au repos).



Disponibilité :Taux de disponibilité attendu :



2.3 Product Backlog

Le Product Backlog en méthodologie Scrum constitue une compilation complète des éléments à développer dans le cadre du projet. Il remplace les documents de spécifications classiques en regroupant à la fois des aspects techniques et fonctionnels, notamment à travers les user stories. Pour notre projet de plateforme de création de chatbots, ce backlog sert de référence évolutive pour guider les développements à venir. Le tableau suivant illustre notre Product Backlog, où les user stories sont classées de 1 à 4 selon leur importance croissante.

ID	Fonctionnalité	ID User Story	Description	Priorité [1-4]	Estimation [J]
1	Authentification	1.1	En tant qu'utilisateur, je veux m'authentifier en sécurité via JWT pour accéder à mon espace personnel.	4	6
		1.2	En tant qu'admin, je veux être notifié lors d'une création d'un nouveau compte et contrôler l'activation du compte	4	4
2	Gestion des utilisateurs & rôles	2.1	En tant qu'administrateur, je veux ajouter ou bannir un utilisateur.	3	4
		2.2	En tant qu'administrateur, je veux gérer les	3	4

			rôles pour contrôler les accès.		
3	Gestion des Templates	3.1	En tant qu'utilisateur, je veux consulter les templates disponibles	3	3
		3.2	En tant qu'administrateur, je veux add, update ou delete (CRUD) des templates.	4	6
4	Création & configuration de chatbot		En tant qu'utilisateur, je veux generer un chatbot avec la template que je veux	4	5
			En tant qu'utilisateur, je veux configurer les paramètres de mon chatbot Et télécharger le code générés	4	5
5	Approvisionnement et déploiement	5.1	En tant qu'utilisateur, je veux que mon chatbot soit déployé automatiquement sur Azure	3	4
		5.2	En tant qu'utilisateur, je veux télécharger les fichiers Terraform de	3	8

			déploiement.		
6	Supervision via Dashboard	6.1	En tant qu'administrateur, je veux consulter les statistiques d'usage via le dashboard.	3	4
		6.2	En tant qu'administrateur, je veux suivre les interactions entre utilisateurs et chatbots.	4	5
		6.3	En tant qu'administrateur, je veux superviser les utilisateurs et leurs activités	3	4

TABLE 2.1 – Product Backlog

2.4 Diagramme de cas d'utilisation Global :

Nous détaillerons ,dans cette partie, les exigences fonctionnelles de notre application à travers le diagramme de cas d'utilisation global.

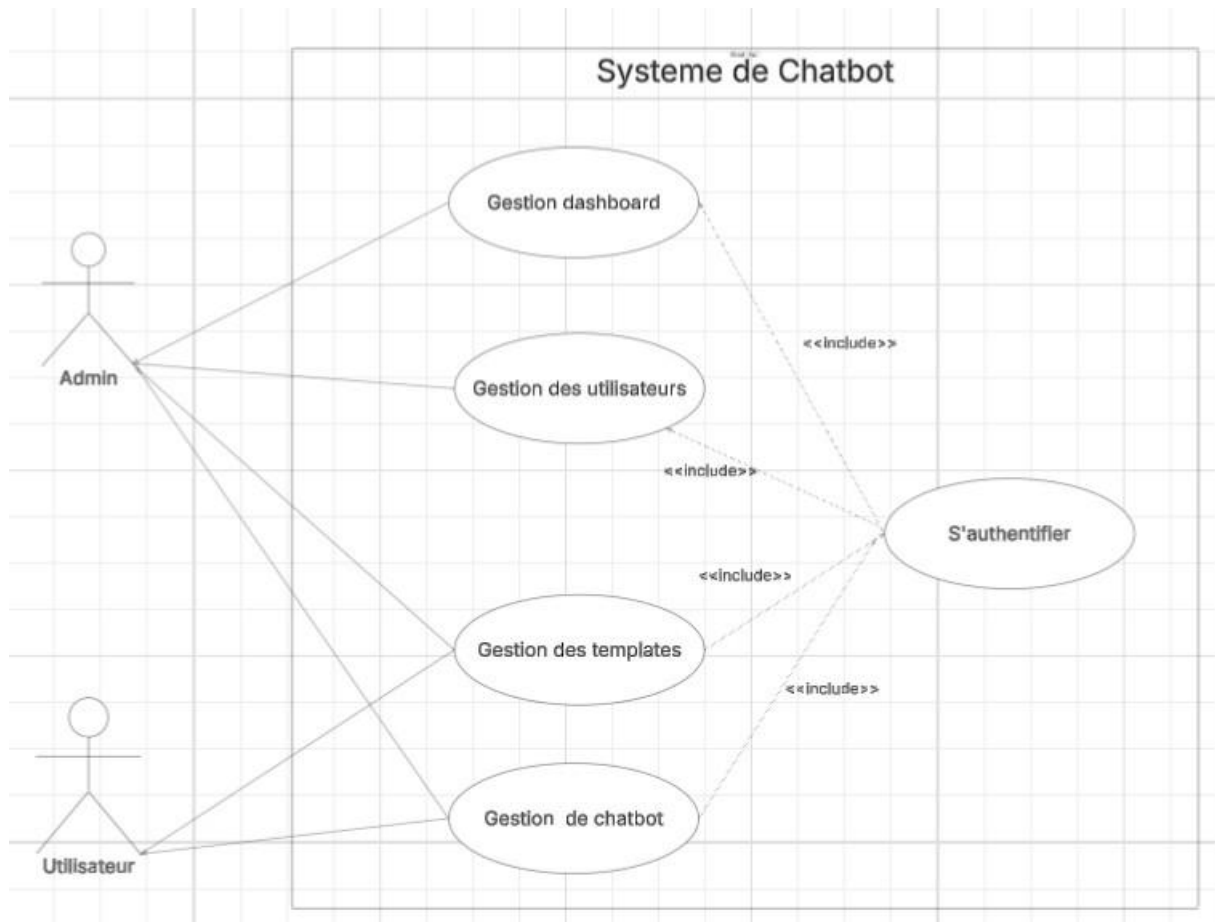


FIGURE2.2 – Diagramme de cas d'utilisation

2.5 Diagramme de Classe :

Dans cette partie, nous exposerons les exigences structurelles de notre application en utilisant le diagramme de classes. Ce diagramme visualise les entités principales du système, leurs propriétés, leurs fonctions, et les relations qui les unissent (associations, héritages, compositions).

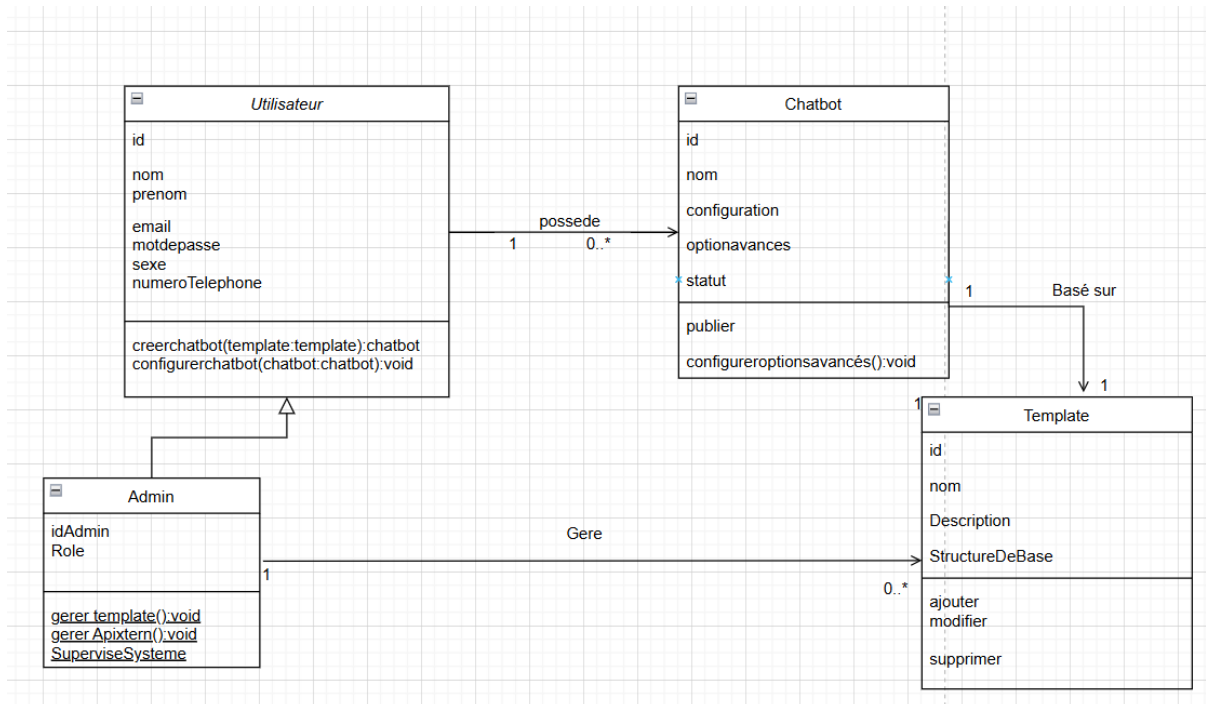


FIGURE 2.3 – Diagramme de classe

2.6 Architecture Globale

2.6.1 Architecture Logique

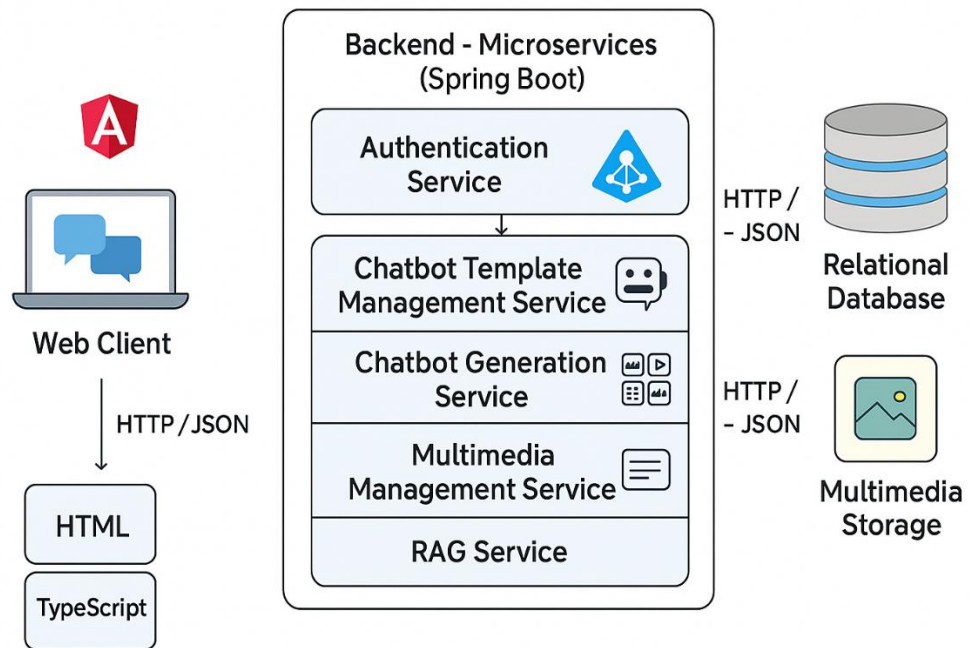


FIGURE 2.4 – Architecture Logique

L'architecture logique de notre plateforme de création de chatbots se base sur une organisation en trois couches:

Couche de présentation : codée avec Angular, elle est chargée de l'affichage de l'interface graphique et de la gestion des interactions utilisateur.

Couche métier : implémentée par des microservices Spring Boot déployés sur des instances Azure App

Services, vise à fournir des services essentiels comme l'authentification (Azure AD),

la gestion et la génération de chatbots, le traitement multimédia ou encore le moteur RAG (Retrieval-Augmented Generation).

Couche d'accès aux données : elle assure la communication entre les services métier et les sources de données, en s'appuyant sur Azure SQL Database pour les données structurées et Azure Blob Storage pour les fichiers multimédias.

2.6.2 Architecture Physique

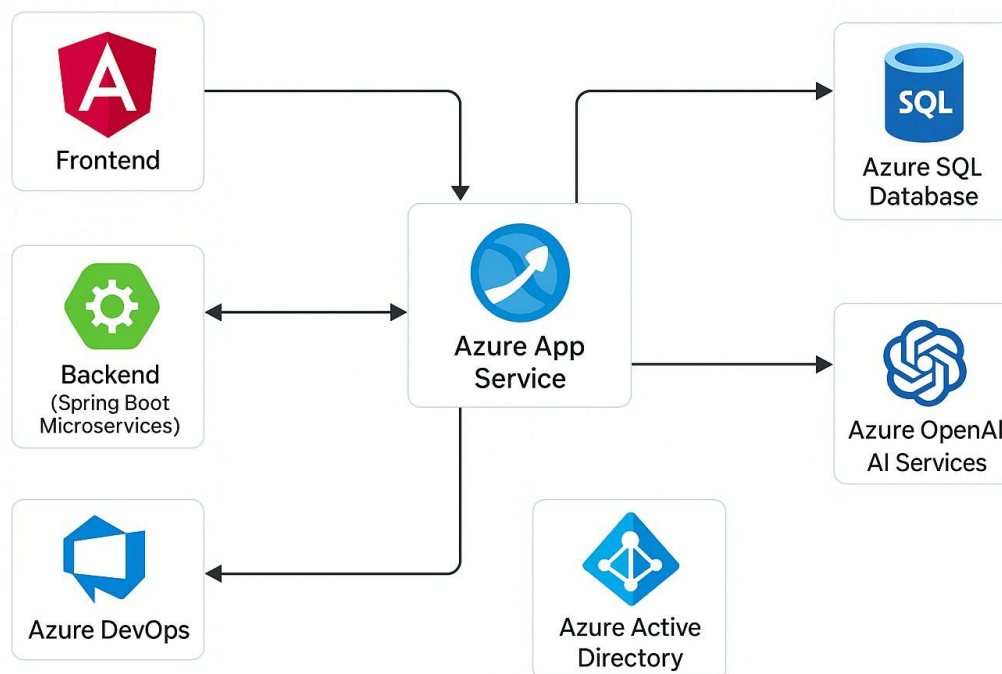


FIGURE 2.5 – Architecture Physique

L'implémentation technique de la plateforme chatbot s'appuie sur une architecture cloud hébergée sur Microsoft Azure, combinant une couche frontend Angular hébergée sur Azure App Service, une couche backend composée de micro-services Spring Boot et également hébergée sur App Service, ainsi que plusieurs services managés.

L'utilisateur échange avec l'interface Angular qui sollicite le backend pour orchestrer les requêtes, gérer les données via Azure SQL Database, ou faire de nouveau appel à l'intelligence artificielle proposée par Azure OpenAI Services

L'authentification et la gestion des droits d'accès sont réalisées par Azure Active Directory (avec support du RBAC), ainsi que l'Intégration Continue et le Déploiement Continu (CI/CD) sont automatisées via Azure DevOps pour un développement agile et sécurisé.

2.7 Environnement de travail

Dans cette partie, nous exposons les instruments de modélisation et de développement employés dans la conception de notre application.

2.7.1 Outils et Technologies utilisés

IntelliJ IDEA	IDE pour le développement du backend en Spring Boot.	
Visual Studio Code	Outil de développement utilisé pour le frontend Angular.	
Azure App Service	Service d'hébergement des applications backend et frontend	
Azure SQL Database	Base de données relationnelle pour stocker les données utilisateurs.	
Azure Active Directory (Entra)	Mécanisme d'authentification et de gestion des rôles (RBAC)	
Azure OpenAI Services	API IA utilisée pour la génération de réponses du chatbot (GPT).	
Azure DevOps	Plateforme de CI/CD et	

(Repos, Pipelines)	gestion du code source	
Angular	Framework côté client employé pour développer l'interface du chatbot	
Spring Boot	Framework backend pour créer des microservices REST sécurisés	
JesonWebToken (auth)	Mécanisme de token utilisé pour authentifier les utilisateurs	
Postman/ GitHub	Postman :environnement de test des APIs REST exposées par le backend. Github : Système d'hébergement du code.	 

Table 2.2 – Les outils et Technologies utilisées

2.8 Planification des sprints

Sprints	Objectif Sprint	Modules concernés	Livrable attendu
Sprint 0	Analyse et spécification des besoins	Tous les modules	-Mise en place de tous les outils nécessaires -Conception du projet finale
Sprint 1	Authentification et gestion des utilisateurs	Gestion des utilisateurs Authentification	-Système d'authentification opérationnel et gestion des roles
Sprint 2	Gestion des chatbots et des templates	Module d'approvisionnement des chatbots	-La partie base de chatbots fonctionnelle -Interface de creation de chatbot
Sprint 3	Développement formulaire dynamique et génération des fichiers terraform	Module d'approvisionnement des chatbots	-Formulaire fonctionnel avec les informations nécessaires à la génération du chatbot
Sprint 4	Développement du dashboard admin	Dashboard Admin	-Dashboard fonctionnel pour gérer les utilisateurs et les chatbots
Sprint 5	Déploiement dynamique sur azure via terraform	Module d'approvisionnement des chatbots	-Déploiement d'une infrastructure chatbot sur azure avec terraform

Table 2.3 – Planification des sprints

2.9 Conclusion

Ce chapitre a permis de présenter le product backlog et de définir les besoins fonctionnels et techniques de la solution les cibles et structures de l'architecture, ainsi que le contexte de la conduite du projet. Nous avons aussi annoncé les principales technologies choisies et les grandes orientations retenues en matière de fiabilité, de sécurité. Nous pouvons désormais rentrer dans le vif du sujet en décrivant le déroulé et le contenu des sprints de développement qui nous conduiront à la mise en production de notre solution.

Chapitre 3

Sprint 1 : Authentification et Gestion des utilisateurs

3.1 Introduction	37
3.2 Analyse et spécification des besoins	37
3.2.1 Sprint Backlog	37
3.3 Diagramme de cas d'utilisation Sprint 1	38
3.4 Diagramme d'activité Sprint 1	39
3.5 Diagramme de séquence système Sprint 1	40
3.6 Conception	41
3.7 Réalisations	42
3.8 Conclusion	45

Sprint 1 : Authentification et Gestion des utilisateurs

3.1 Introduction

Avec ce sprint, nous allons explorer comment gérer les utilisateurs et les rôles, un volet essentiel pour protéger l'accès à notre plateforme de création de chatbots. Nos débuts seront méthodiques : nous énumérerons les tâches à réaliser en les estimant dans notre Sprint Backlog viendra ensuite la conception de ce module (structures des entités, rôles, permissions) suivie de sa réalisation (authentification, attribution des rôles, gestion des accès).

En conclusion, nous ouvrirons une phase de test pour être sûrs de la fiabilité et de la sécurité de notre système.

3.2 Analyse et spécification des besoins

3.2.1 Sprint backlog

ID	Fonctionnalités	ID User Story	Description	Priorité [1-4]	Estimation
1	Authentification	1.1	En tant qu'utilisateur/admin je veux m'authentifier	4	4
		1.2	En tant qu'utilisateur, je veux une authentification sécurisée via JWT.	4	4
2	Gestion des utilisateurs & rôles	2.1	En tant qu'administrateur, je veux ajouter ou bannir un utilisateur.	3	4
		2.2	En tant qu'administrateur, je veux gérer les rôles	3	4

Table 2.4 Sprint Backlog du Sprint 1

3.3 Diagramme de cas d'utilisation de sprint 1

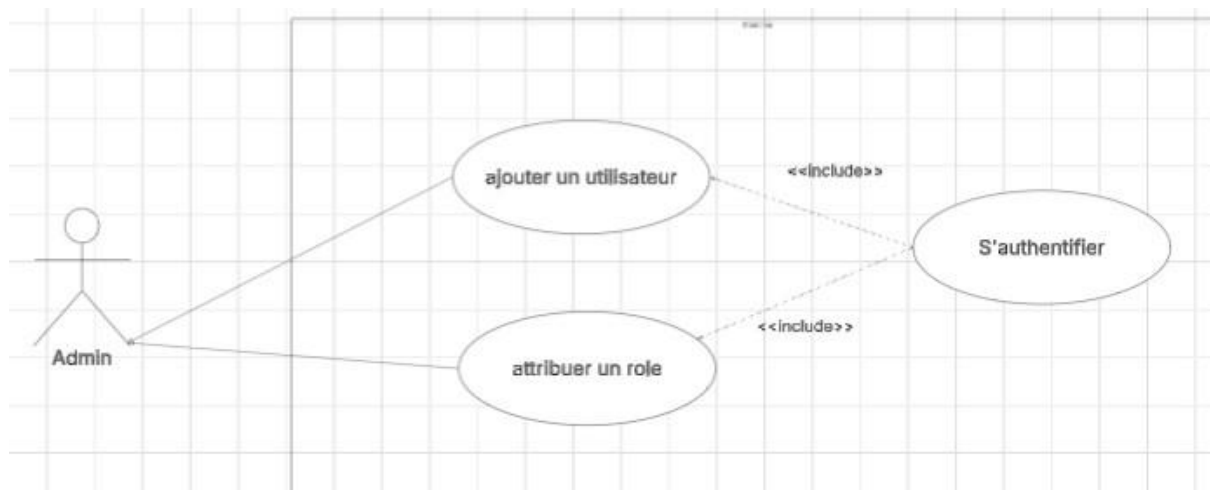


FIGURE 3.1 – Diagramme de cas d'utilisation de sprint 1

- Énoncé textuel du cas d'utilisation intitulé « Procédure d'Authentification »

Cas d'utilisation	Authentification
Acteur	Admin / Utilisateur
Description	-les acteurs qui sont ajoutés par l'admin peuvent s'authentifier avec ce cas d'utilisation
Postcondition	-vous êtes authentifié avec succès
Scénario Nominal	-l'admin ajoute les rôles -il ajoute un utilisateur en lui donnant un rôle à la deuxième étape -l'utilisateur ajouter peut se connecter avec son nom d'utilisateur et son mot de passe qui sont valides
Scénario Alternative	-si un champ vide mot de passe ou non d'utilisateur -alerte par le système

Table 2.5 – Procédure d'Authentification

- Présentation verbale du cas d'utilisation nommé « Ajouter Utilisateur »

Cas d'utilisation	ajouter un utilisateur
Acteur	Admin
Description	l'admin peut contrôler les informations des utilisateurs avec ce cas d'utilisation
Postcondition	l'admin affecte un rôle pour l'utilisateur ajouté
Scenario Nominal	-l'utilisateur clique sur Ajouter utilisateur -un formulaire d'ajout se recharge -l'utilisateur remplit les champs -l'utilisateur clique sur register -l'admin clique sur Ajouter en dashboard -message succès afficher
Scenario Alternative	-un champ vide -une alerte afficher sur l'écran

Table 2.6 – Ajouter Utilisateur

3.4 Diagramme d'activité Création d'un nouveau compte

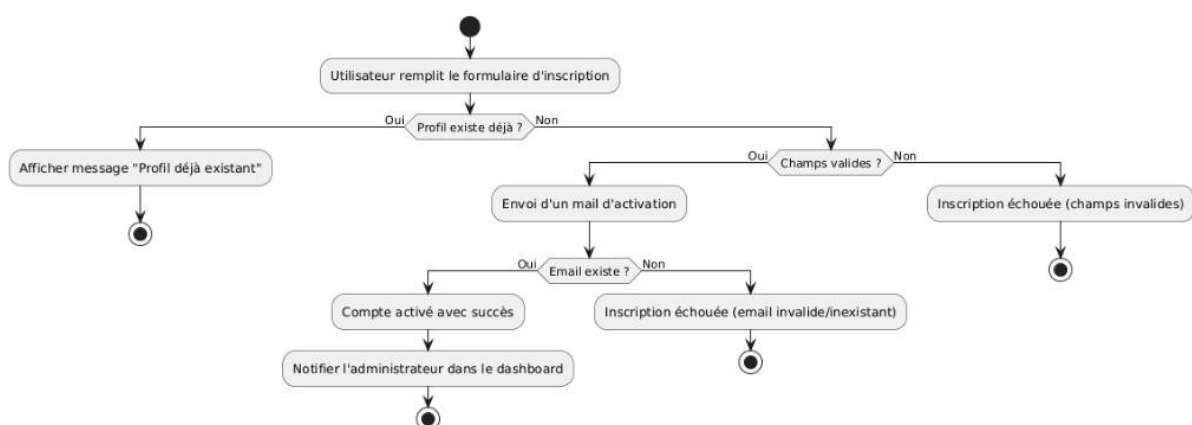


Figure 3.2-Diagramme d'activité « Création d'un nouveau compte

3.5 Diagramme de Séquence Système

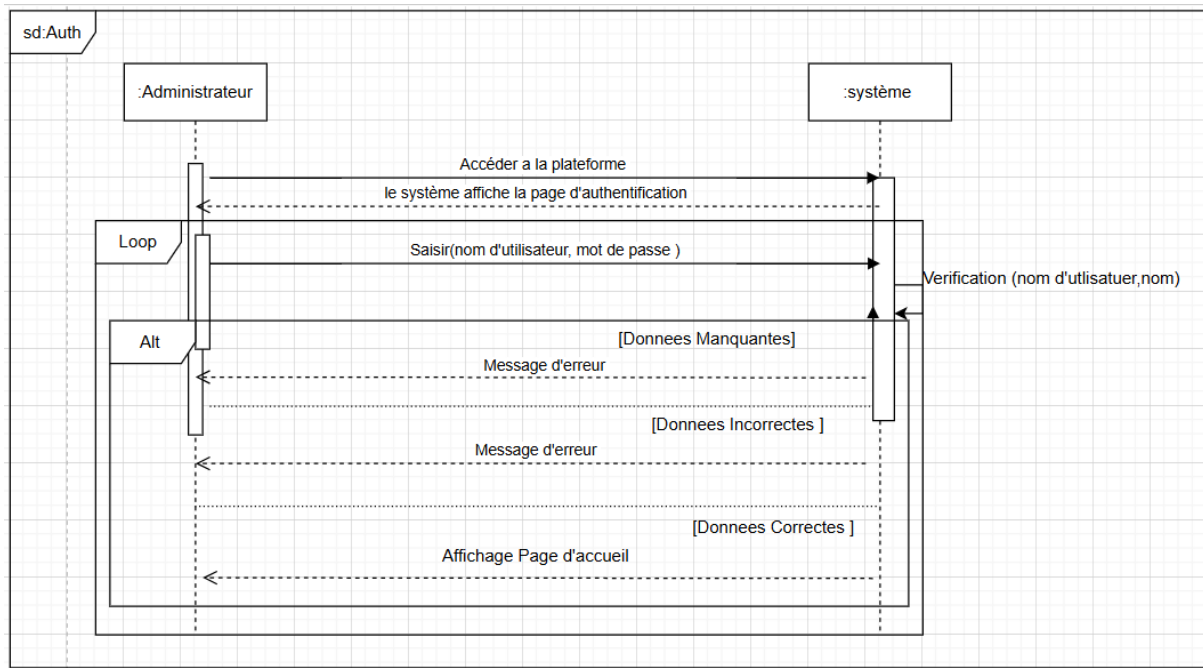


FIGURE 3.3 – Diagramme Séquence Système

3.6 Conception

Dans cette section, nous présentons deux diagrammes de séquence objet pour illustrer les interactions liées à la sécurité, notamment l'authentification et la gestion des rôles. Ces diagrammes montrent comment les objets du système interagissent dans le temps, en mettant l'accent sur la circulation des messages entre les composants. La figure ci-dessous illustre le diagramme de séquence objet « Sign In », représentant le déroulement de l'authentification d'un utilisateur.

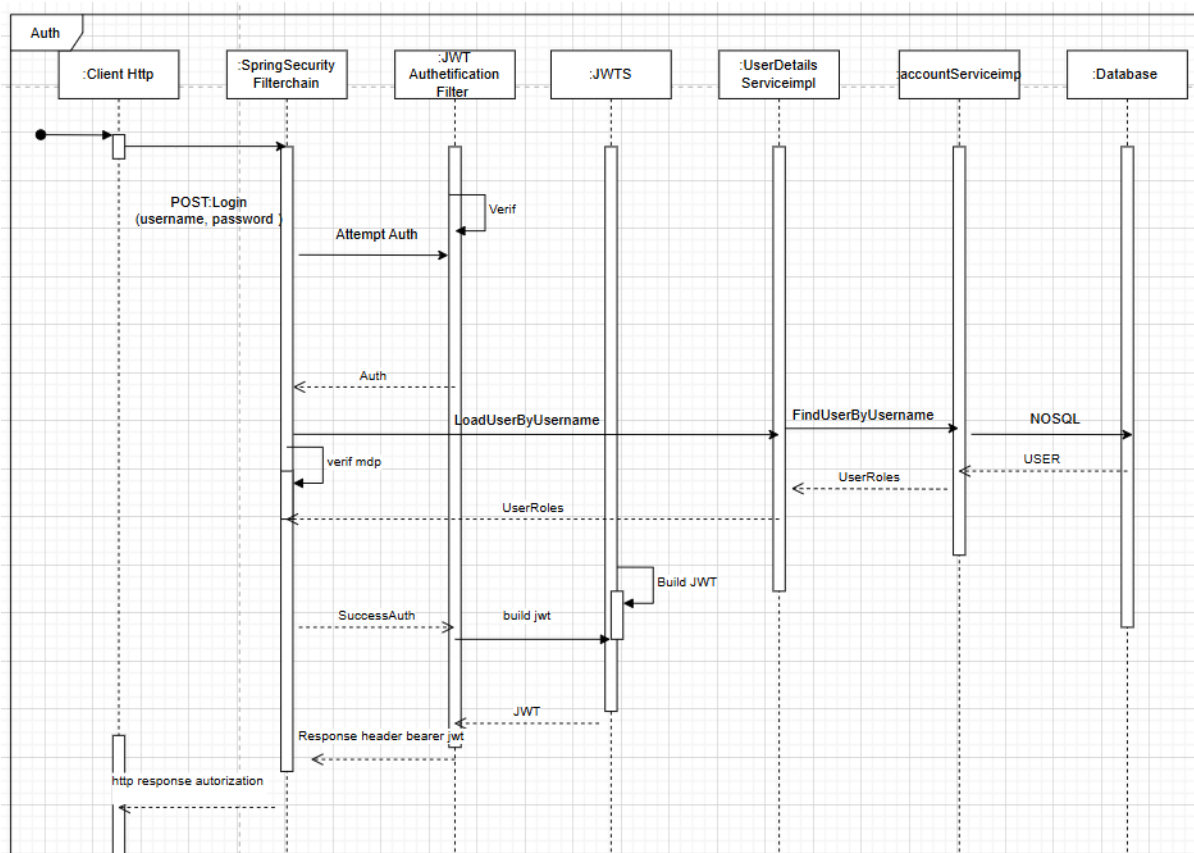


FIGURE 3.4 – Diagramme de séquence objet du cas d'utilisation «Se Connecter»

3.7 Réalisations

Dans cette partie, nous illustrerons quelques interfaces du système de notre sprint

1.

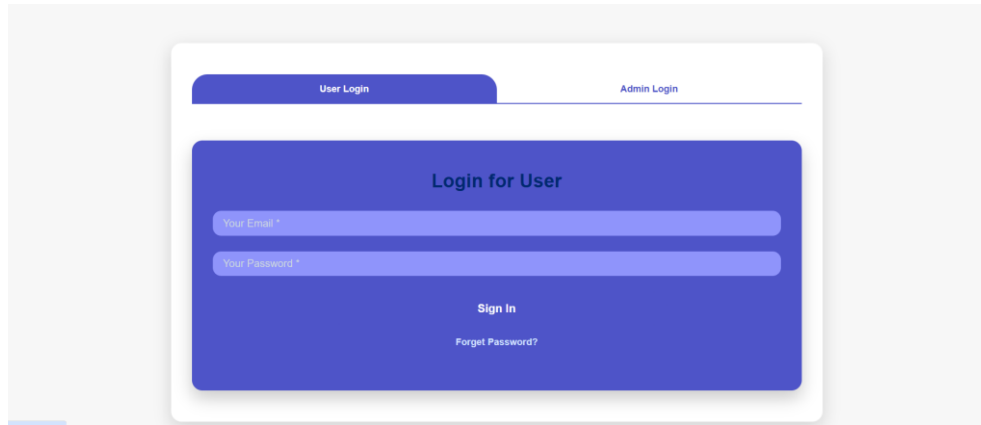


FIGURE 3.5 – Authentification

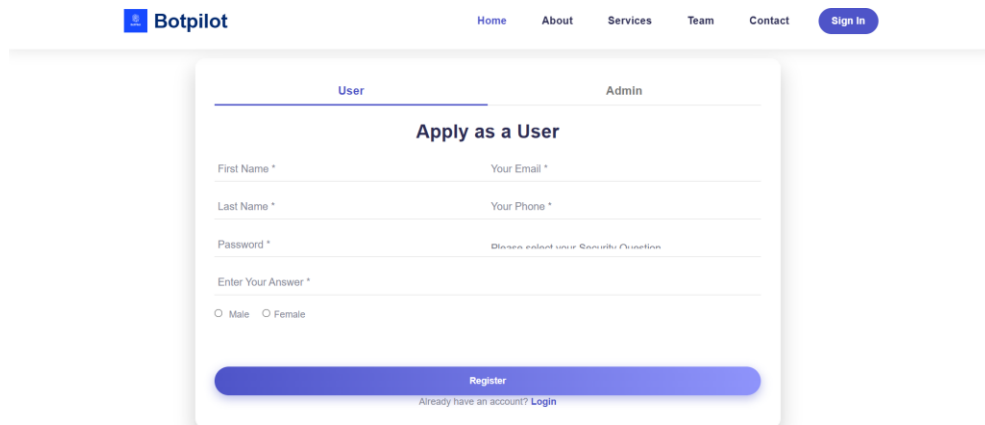


FIGURE 3.6 – Créer un compte

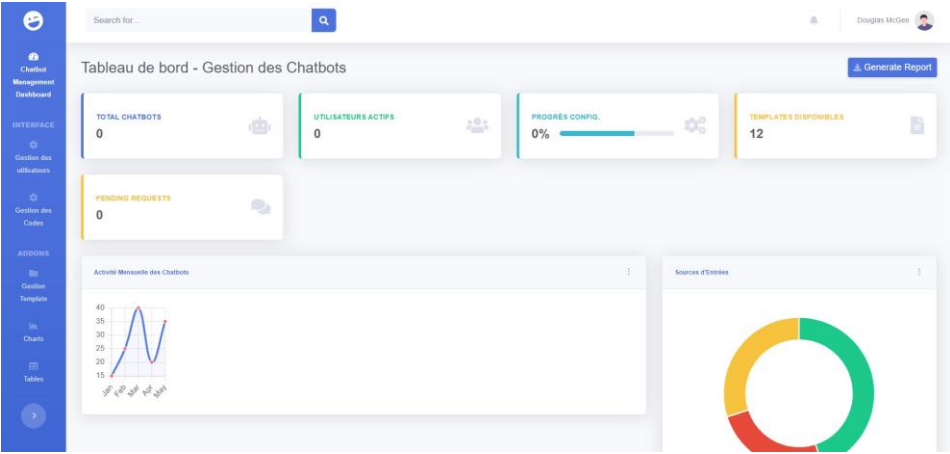


FIGURE 3.7– Dashboard Admin

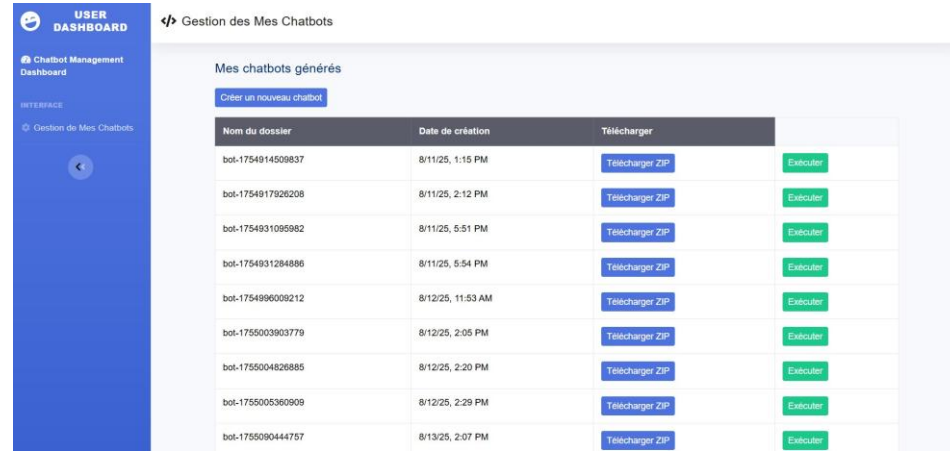


FIGURE 3.8– Dashboard User

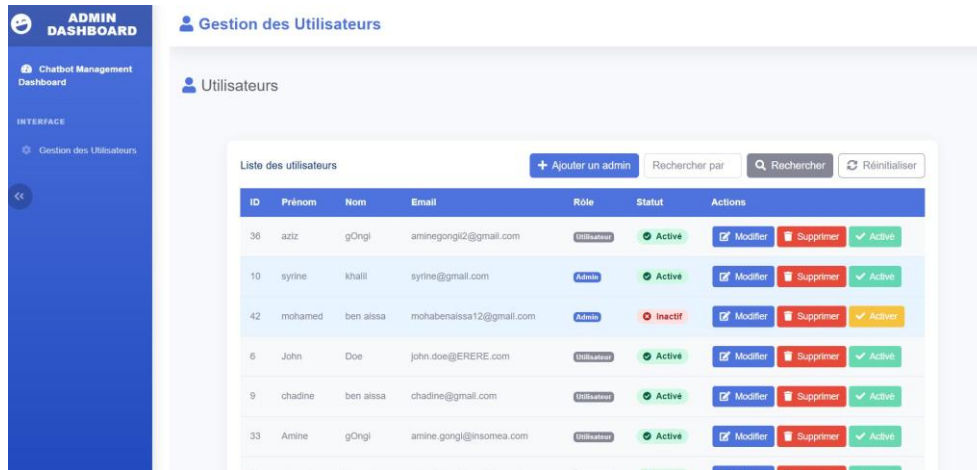


FIGURE 3.9 – Dashboard Admin : Gestion des utilisateurs

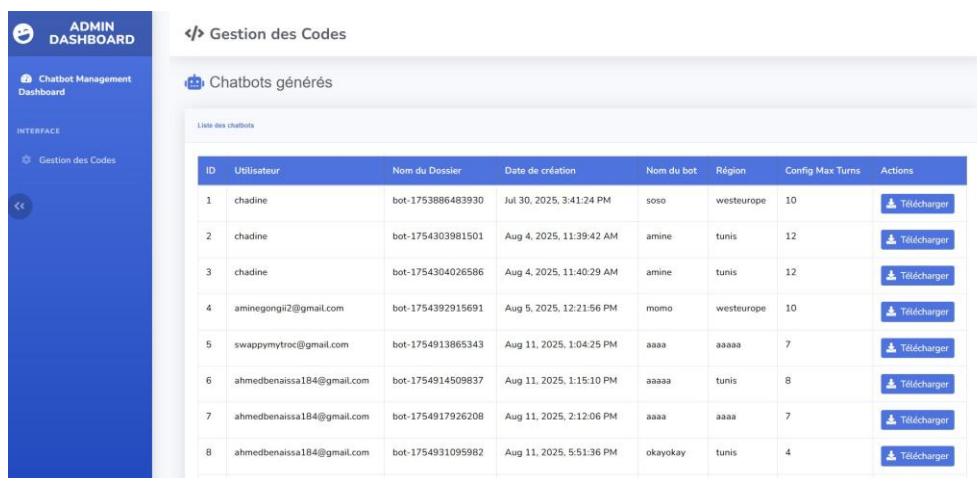


FIGURE 3.10– Dashboard Admin : Gestion des codes

3.8 Conclusion

Dans cette partie, nous avons développé la gestion des utilisateurs et des rôles pour notre plateforme de création de chatbots. Deux rôles ont été définis : administrateur, chargé de gérer les templates et les utilisateurs, et utilisateur standard, limité à la création de chatbots. Cette distinction permet un meilleur contrôle des accès et une utilisation sécurisée de la plateforme

Chapitre 4

Sprint 2 : Gestion des Chatbots et des Templates

4.1 Introduction	47
4.2 Analyse et spécification des besoins	47
4.2.1 Sprint Backlog	47
4.3 Diagramme de cas d'utilisation Sprint 2	48
4.4 Diagramme de séquence Sprint 2	48
4.5 Réalisations	49
4.7 Conclusion	49

Sprint 2 : Gestion des Chatbots et des Templates

4.1 Introduction

Après avoir traité le sprint "Gestion des utilisateurs et des rôles", nous allons nous intéresser à la gestion des sprints du "Gestion des templates et des chatbots", qui consiste à développer la partie standard du chatbot, socle commun pour tous les chatbots à venir. Nous verrons tout d'abord le sprint backlog, c'est-à-dire l'ensemble de fonctionnalités attendues. Nous verrons ensuite comment étudier et concevoir ce composant applicatif, puis comment il a été réalisé et contrôlé.

4.2 Analyse et spécification des besoins

4.2.1 Sprint Backlog :

ID	Fonctionnalités	ID User Story	Description	Priorité [1-4]	Estimation
1	Gestion des templates	1.1	En tant qu'utilisateur, je veux accéder aux templates disponibles	3	3
		1.2	En tant qu'administrateur, je veux add, update ou delete (CRUD) des templates	4	6
2	Gestion de création de chatbot	2.1	En tant qu'utilisateur, je veux generer un chatbot avec Rag	4	5
		2.2	En tant qu'utilisateur, je veux configurer les paramètres de mon chatbot.	4	5

Table 3.1 -Sprint Backlog du Sprint 2

4.3 Diagramme de cas d'utilisation Sprint 2

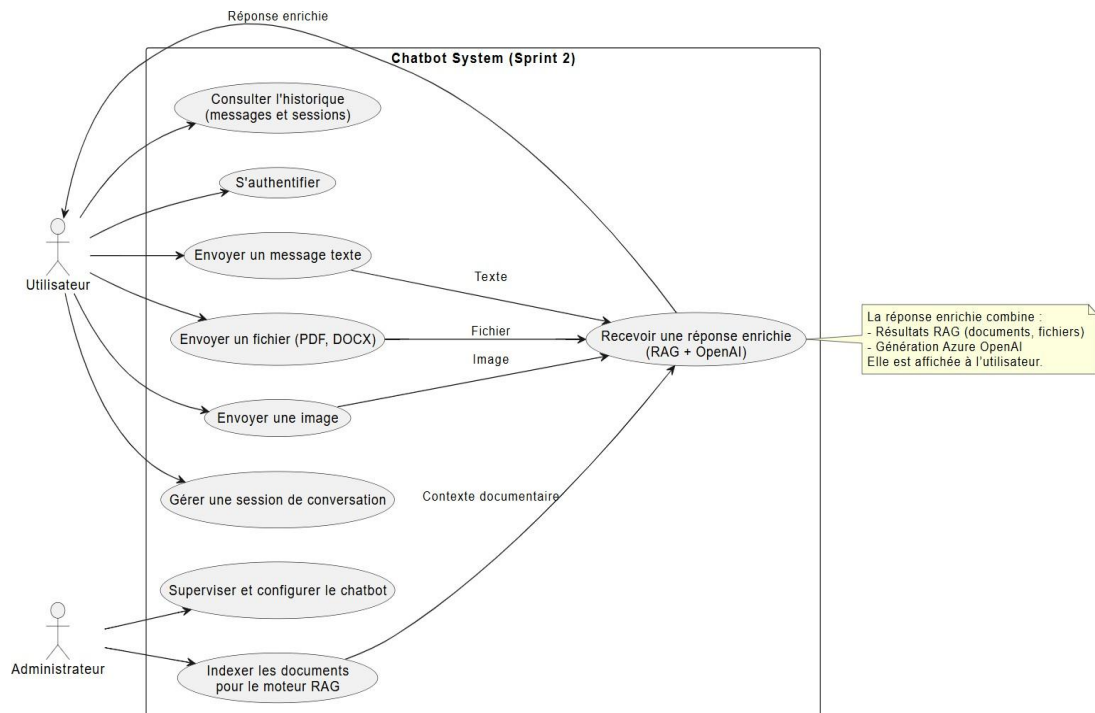


FIGURE 4.1 – Diagramme de cas d'utilisation Sprint 2

4.4 Diagramme de séquence objet Sprint 2

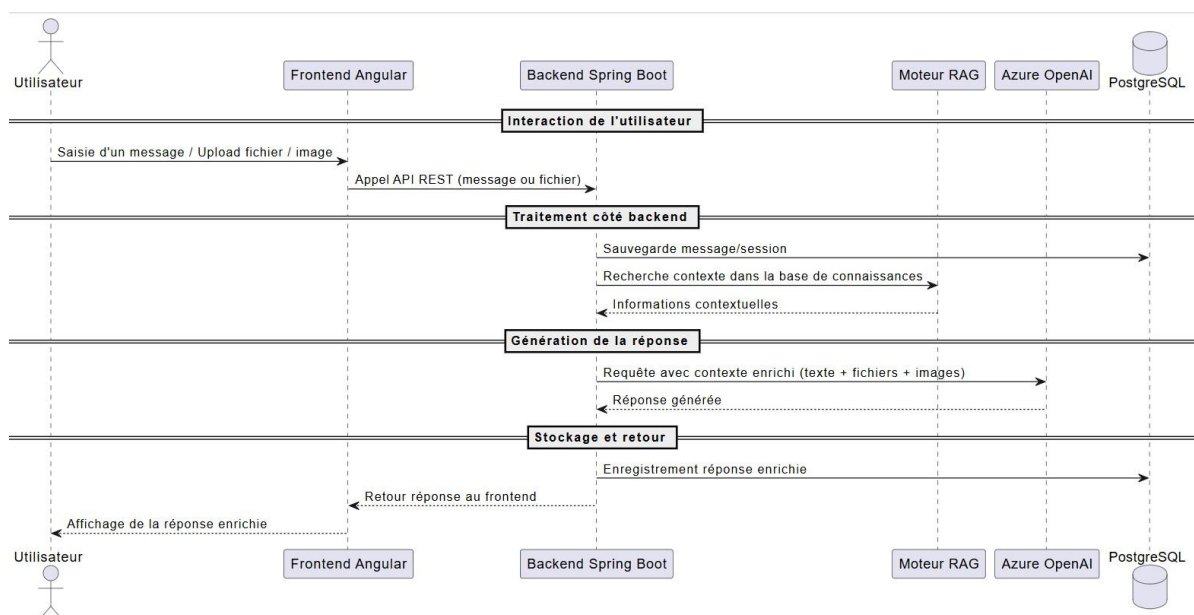


FIGURE 4.2 – Diagramme de séquence objet de Sprint 2

4.5 Réalisations

Développement de la partie base de chatbot :



FIGURE 4.3 – Premier chatbot réalisé avec Rag

4.6 Conclusion :

À l'achèvement de ce sprint :

- **Le chatbot de base est opérationnel** : il reçoit des entrées texte, fichiers et images, et répond en s'appuyant sur Azure OpenAI + moteur RAG.
- **Le moteur RAG est intégré** : les documents et images sont prétraités, indexés et consultés via des embeddings vectoriels. Le chatbot exploite ces résultats contextuels pour enrichir ses réponses.
- **La multimodalité est supportée** : l'utilisateur peut interagir non seulement en texte mais aussi via fichiers (ex : PDF, DOCX) et images.

Le socle est réutilisable et extensible, prêt à intégrer : des formulaires dynamiques (Sprint 3) et un pipeline Terraform pour automatiser le déploiement

Chapitre 5

Sprint 3 : Formulaire Dynamique et Génération des Fichiers Terraform

5.1 Introduction	51
5.2 Analyse et spécification des besoins	51
5.2.1 Les Champs Du Formulaire	51
5.2.2 Les Besoins Utilisateurs	52
5.2.3 Les Besoins Administrateur	52
5.2.4 Sprint Backlog	53
5.3 Diagramme de cas d'utilisation et de Séquence	54
5.3.1 Diagramme de cas d'utilisation Sprint 3	54
5.3.2 Diagramme de séquence Sprint 3	55
5.4 Génération des Fichiers Terraform	56
5.4.1 Objectif	56
5.4.2 Les Composants Générés	56
5.4.3 La Logique de Génération	56
5.4.4 Les Points Importants	56
5.5 Réalisations	57
5.6 Conclusion	59

Sprint 3 : Formulaire Dynamique et Génération des Fichiers Terraform

5.1 Introduction

Ce sprint vise à permettre à l'utilisateur de configurer un chatbot via un formulaire complet et de créer les fichiers de manière automatisée Terraform pour le déploiement sur Azure.

5.2 Analyse et spécification des besoins

5.2.1 Champs du formulaire :

1. **Nom du bot** : Nom unique donné au chatbot pour l'identifier.
2. **Nom du groupe de ressources** : Nom du groupe Azure où le chatbot sera déployé, utilisé pour organiser les ressources.
3. **Template** : Style visuel du chatbot que l'utilisateur choisit parmi : Clair, Sombre, Colorful.
4. **SKU (Stock Keeping Unit)** : Permet de définir le niveau de service et les fonctionnalités du chatbot. Les options sont : Basic, Standard, Premium.
 - **Basic** : Fonctionnalités limitées, adapté aux petits projets ou tests. Max Turns : 10, Timeout : 30 secondes.
 - **Standard** : Fonctionnalités intermédiaires, adapté aux projets moyens. Max Turns : 50, Timeout : 60 secondes.
 - **Premium** : Fonctionnalités complètes, adapté aux projets avancés. Max Turns : 100, Timeout : 120 secondes.
5. **Size** : Détermine la taille de l'instance du chatbot. Influence la capacité de traitement et la performance.
 - **Small** : Faible capacité, adapté aux environnements de test.
 - **Medium** : Capacité moyenne, adapté aux projets moyens.
 - **Large** : Haute capacité, adapté aux projets à forte charge.
6. **Max Turns** : Nombre maximal de tours de conversation autorisés par session. Limite la longueur d'une interaction.
 - Basic : 10 tours
 - Standard : 50 tours
 - Premium : 100 tours
7. **Timeout** : Durée maximale d'inactivité avant fermeture de la session de chat (en secondes).
 - Basic : 30 secondes

- Standard : 60 secondes
- Premium : 120 secondes

5.2.2 Les Besoins Utilisateur :

- Remplir le formulaire complet avec tous les champs requis.
- Soumettre le formulaire pour déclencher la génération des fichiers Terraform et du code du chatbot.

5.2.3 Les Besoins Administrateur :

- Gérer les templates disponibles pour le champ Template.
- Recevoir des notifications lorsqu'un utilisateur génère un chatbot.

5.2.4 Sprint Backlog :

ID Fonctionnalité	ID User Story	Description	Priorité [1-4]	Estimation [J]
1	1.1	En tant qu'utilisateur, je veux remplir un formulaire complet pour configurer mon chatbot	1	2
1	1.2	Je veux choisir une template (Clair, Sombre, Colorful) pour mon chatbot	2	1
2	2.1	Je veux sélectionner un SKU (Basic, Standard, Premium) pour définir les fonctionnalités du chatbot	1	1
2	2.2	Je veux définir la taille (Small, Medium, Large) de l'instance pour gérer la capacité	2	1
1	1.3	Je veux soumettre le formulaire pour générer automatiquement les fichiers Terraform et le code du chatbot	1	2
3	3.1	En tant qu'administrateur, je veux recevoir des notifications lorsqu'un utilisateur génère un chatbot	2	1

ID Fonctionnalité	ID User Story	Description	Priorité [1-4]	Estimation [J]
3	3.2	Je veux gérer les templates disponibles pour que les utilisateurs aient des options à jour	3	2
4	4.2	Je veux générer automatiquement les fichiers Terraform pour faciliter le déploiement sur Azure	1	2

Table 4.1 – Sprint Backlog du Sprint 3

5.3 Diagrammes de cas d'utilisation et de séquence

5.3.1 Diagramme de cas d'utilisation :

Ce diagramme illustre les fonctionnalités du formulaire de configuration et de génération de chatbots, ainsi que les interactions entre l'utilisateur et l'administrateur

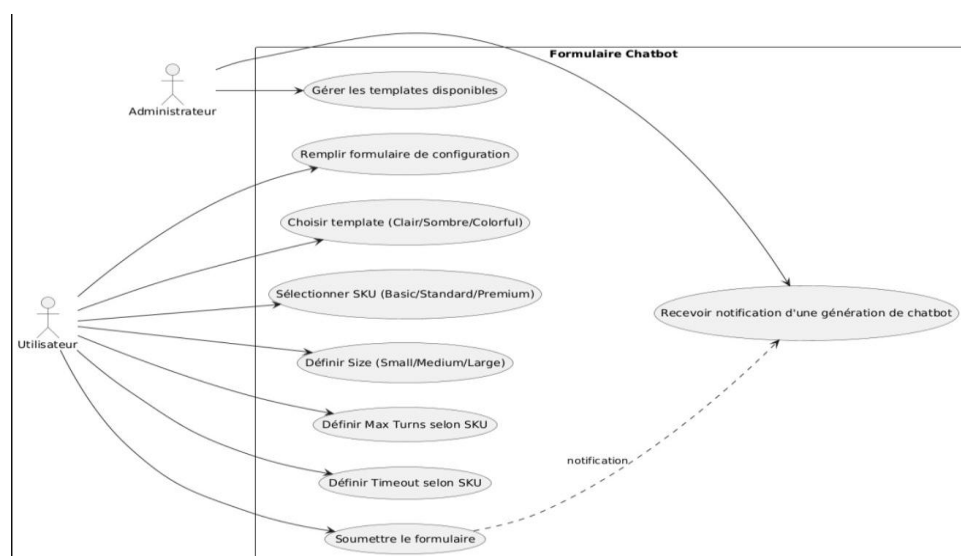


Figure 5.1 – Diagramme de cas d'utilisation Sprint 3

5.3.2 Diagramme de Séquence Sprint 3 :

Ce diagramme montre le processus de génération d'un chatbot à partir du formulaire de configuration. Il illustre le déroulement des interactions entre l'utilisateur, le frontend et le backend, la génération des fichiers Terraform et l'administrateur, en différenciant les scénarios de réussite et d'erreur.

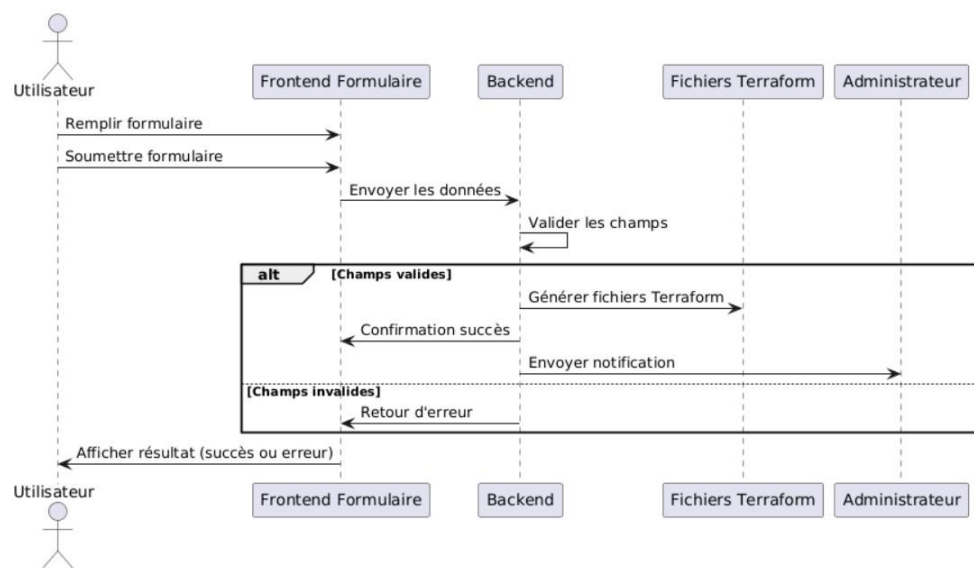


FIGURE 5.2 – Diagramme de séquence Sprint 3

5.4 Génération Des Fichiers Terraform

5.4.1 Objectif :

L'objectif de cette partie est de gérer de manière automatisée le déploiement des ressources Azure nécessaires au chatbot .La génération des infrastructures doit se faire sans traitement manuel garantissant la répétabilité et la cohérence des environnements créés.

5.4.2 Les Composants Générés :

Les principaux composants générés par Terraform sont :

Resource Group : conteneur principal pour toutes les ressources du chatbot.

App Service : hébergement de l'application chatbot.

Blob Storage : stockage des fichiers et données nécessaires.

Key Vault : stockage sécurisé des secrets et clés.

Autres ressources Azure : selon la configuration choisie dans le formulaire (SKU, Size, Max Turns, Timeout).

5.4.3 La Logique de Génération :

Chaque soumission de formulaire déclenche la création dynamique d'un fichier Terraform adapté aux options choisies (template, SKU, Size, etc.). Les fichiers contiennent les ressources Azure configurées selon les choix de l'utilisateur. Une validation des données est effectuée avant génération pour s'assurer de leur cohérence. Une fois les fichiers générés, ils peuvent être déployés directement sur Azure.

5.4.4 Les Points Importants :

- Structure du projet Terraform généré : dossier, fichiers principaux ('main.tf', 'variables.tf', 'outputs.tf').
- Gestion des variables dynamiques provenant du formulaire.
- Gestion des erreurs et des cas où la génération échoue.
- Notification à l'administrateur après création des fichiers.

5.5 Réalisations :

Étape 1

Étape 2

Étape 3

Informations de base

Nom du bot *

Resource Group *

Suivant

Étape 1

Étape 2

Étape 3

Configuration

Région *

Clair

-- Sélectionner SKU Tier --

-- Sélectionner SKU Size --

Max Turns *

Timeout (sec) *

Étape 1

Étape 2

Étape 3

Message de bienvenue

Message de bienvenue

Précédent Créer le bot

FIGURE 5.3 – Formulaire de configuration

Nom	Modifié le	Type	Taille
 bot-1751543639237	03/07/2025 12:53	Dossier de fichiers	
 bot-1751544817616	03/07/2025 13:13	Dossier de fichiers	
 bot-1751545410262	03/07/2025 13:23	Dossier de fichiers	
 bot-1751545417157	03/07/2025 13:23	Dossier de fichiers	
 bot-1751545736647	03/07/2025 13:28	Dossier de fichiers	
 bot-1751546828084	03/07/2025 13:47	Dossier de fichiers	
 bot-1751547081502	03/07/2025 13:51	Dossier de fichiers	
 bot-1751547318375	03/07/2025 13:55	Dossier de fichiers	
 bot-1751547518531	03/07/2025 13:58	Dossier de fichiers	
 bot-1751547736716	03/07/2025 14:02	Dossier de fichiers	
 bot-1751548985123	03/07/2025 14:23	Dossier de fichiers	
 bot-1751549243104	03/07/2025 14:27	Dossier de fichiers	
 bot-1751549747573	03/07/2025 14:35	Dossier de fichiers	
 bot-1751549773684	03/07/2025 14:36	Dossier de fichiers	
 bot-1751550241727	03/07/2025 14:44	Dossier de fichiers	
 bot-1751551335666	03/07/2025 15:02	Dossier de fichiers	
 bot-1752066347654	09/07/2025 14:05	Dossier de fichiers	
 bot-1752067640335	09/07/2025 14:27	Dossier de fichiers	
 bot-1752068135242	09/07/2025 14:35	Dossier de fichiers	
 bot-1752068701117	09/07/2025 14:17	Dossier de fichiers	

FIGURE 5.4 – les chatbots créés injectés automatiquement dans des dossiers

```
{
  "botName": "ahmedbot",
  "region": "westeurope",
  "resourceGroup": "rgahmed",
  "sku": {
    "tier": "Standard",
    "size": "S1"
  },
  "maxTurns": "10",
  "timeout": "2",
  "welcomeMessage": "hello how can i help you "
}
```

FIGURE 5.5 – la Template Terraform du formulaire

5.6 Conclusion

Au terme de ce sprint, nous avons conçu et implémenté le formulaire dynamique de configuration du chatbot, intégrant le choix du template, du SKU, de la taille, ainsi que des paramètres spécifiques tels que Max Turns et Timeout. Nous avons également mis en place la génération automatisée des fichiers Terraform, permettant de préparer toutes les ressources nécessaires au déploiement sur Azure de manière fiable et répétable.

Cette étape a permis de valider la cohérence des données fournies par l'utilisateur et d'assurer une coordination efficace entre le frontend, le backend et l'infrastructure Terraform. Les notifications destinées à l'administrateur garantissent un suivi efficace des créations de chatbots et facilitent la gestion des templates disponibles.

Ainsi, ce sprint pose les bases solides pour le prochain chapitre, dédié au **Déploiement dynamique sur Azure via Terraform**, où nous détaillerons comment ces fichiers générés sont utilisés pour créer automatiquement l'infrastructure complète du chatbot dans le cloud.

Chapitre 6

Sprint 4 : Déploiement dynamique sur azure via Terraform

6.1 Introduction	61
6.2 Analyse et spécification des besoins	61
6.2.1 Sprint Backlog	61
6.3 Diagramme de cas d'utilisation et de Séquence	63
6.3.1 Diagramme de cas d'utilisation Sprint 4	63
6.3.2 Diagramme de séquence Sprint 4	64
6.4 Formulaire de Déploiement	64
6.4.1 Objectif	64
6.4.2 Les Champs du Formulaire	64
6.4.3 La Logique de Validation	64
6.4.4 Les Points Importants	65
6.5 Réalisations	66
6.6 Conclusion	66

Sprint 4 : Déploiement dynamique sur azure via Terraform

6.1 Introduction

Le but de ce sprint est d'introduire le déploiement automatique des chatbots sur Azure à partir des fichiers Terraform générés lors du sprint précédent. Les besoins identifiés sont les suivants :

- Déployer de manière automatisée les ressources Azure nécessaires pour chaque chatbot.
- Assurer la cohérence et la répétabilité du déploiement.
- Gérer les erreurs et les exceptions lors du déploiement.
- Permettre à l'administrateur de suivre les déploiements et de recevoir des notifications en cas de succès ou d'échec.
- Garantir que le processus respecte les choix de configuration de l'utilisateur (template, SKU, Size, Max Turns, Timeout).

6.2 Analyse et spécification des besoins

6.2.1 Sprint Backlog :

ID Fonctionnalité	ID User Story	Description	Priorité [1-4]	Estimation [J]
1	1.1	En tant qu'utilisateur, je veux remplir le formulaire de déploiement avec mes credentials Azure (Client ID, Client Secret, Tenant ID, ID d'abonnement) pour que le déploiement se fasse dans mon propre abonnement	1	2
	1.2	En tant qu'utilisateur, je souhaite lancer le déploiement des fichiers Terraform générés sur Azure en utilisant mes identifiants.	1	2
	1.3	En tant qu'utilisateur, je veux que les variables dynamiques du formulaire (template, SKU, Size, Max Turns, Timeout) soient prises en compte lors du déploiement	1	1

2	2.1	En tant qu'utilisateur, je veux vérifier que les ressources Azure ont été créées correctement dans mon abonnement	1	2
	2.2	En tant qu'utilisateur, je veux que les erreurs et exceptions du déploiement soient gérées et affichées clairement	1	2
3	3.1	En tant qu'administrateur, je veux recevoir une notification indiquant le succès ou l'échec du déploiement pour suivre les chatbots créés	2	1
	3.2	En tant qu'administrateur, je veux suivre le statut des déploiements via le dashboard	2	1
4	4.1	En tant qu'utilisateur, je veux tester le processus complet de déploiement pour différents scénarios afin de m'assurer de sa fiabilité	2	2
	4.2	En tant qu'administrateur, je veux documenter le processus de déploiement pour assurer sa cohérence et faciliter la maintenance	3	1

Table 5.1 – Sprint Backlog du Sprint 4

6.3 Diagrammes de cas d'utilisation et de séquence

Cette section présente les diagrammes permettant de visualiser les échanges entre les acteurs et le système pour le Sprint 4. Ils montrent comment l'utilisateur et l'administrateur interagissent avec le formulaire de déploiement et le processus Terraform, en illustrant à la fois les fonctionnalités disponibles et le flux des opérations.

6.3.1 Diagramme de cas d'utilisation de Sprint 4

Le diagramme de cas d'utilisation illustre l'ensemble des fonctionnalités disponibles dans le Sprint 4, depuis le remplissage du formulaire de déploiement jusqu'à la notification de l'administrateur. Il permet de visualiser les rôles de chaque acteur et leurs échanges avec le système.



FIGURE 6.1 – Diagramme de cas d'utilisation Sprint 4

6.3.2 Diagramme de séquence de Sprint 4

Le diagramme de séquence décrit le flux des opérations lors du déploiement dynamique sur Azure. Il décrit le déroulement des interactions entre l'utilisateur, le frontend, le backend, Terraform et l'administrateur, en précisant la gestion des succès et des échecs lors du déploiement.

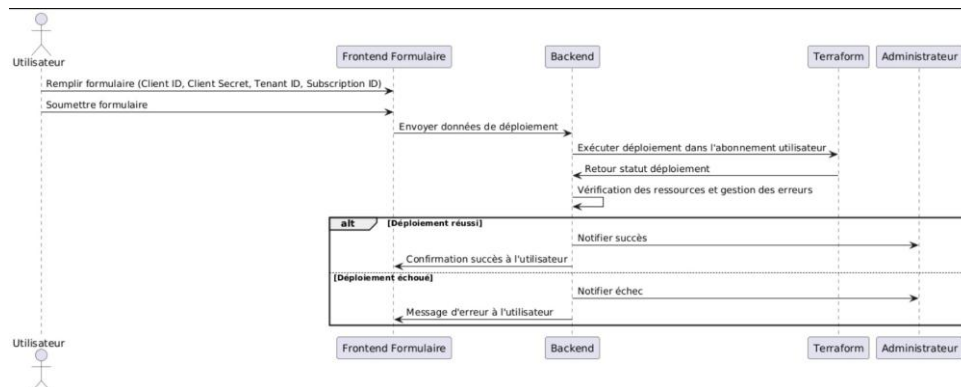


FIGURE 6.2 – Diagramme de séquence Sprint 4

6.4 Formulaire de déploiement

6.4.1 Objectif

L'objectif de cette partie est de permettre à l'utilisateur de déployer le chatbot dans son propre abonnement Azure. Le formulaire englobe toutes les informations nécessaires pour générer et exécuter les fichiers Terraform, garantissant la cohérence des données et facilitant le processus automatisé de déploiement.

6.4.2 Champs du formulaire

Le formulaire contient les champs suivants :

Client ID : identifiant de l'application Azure Active Directory.

Client Secret : clé secrète pour l'authentification.

Tenant ID : identifiant du tenant Azure.

Subscription ID : identifiant de l'abonnement où les ressources seront créées.

6.4.3 La Logique de validation :

- Vérification de la validité des **crédenciales Azure**.
- Validation des champs obligatoires
- Vérification des valeurs numériques

6.4.4 Les Points importants :

- Les données saisies sont utilisées directement pour générer les fichiers Terraform.
- Chaque utilisateur peut déployer dans son propre abonnement sans interférer avec les autres.
- Gestion des erreurs si les credentials sont invalides ou si le déploiement échoue.
- Notifications envoyées à l'utilisateur et à l'administrateur après l'exécution du déploiement.

6.5 Réalisations :

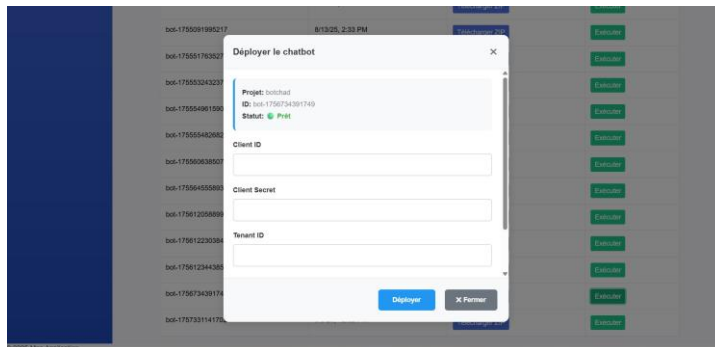


FIGURE 6.3 – Formulaire de déploiement

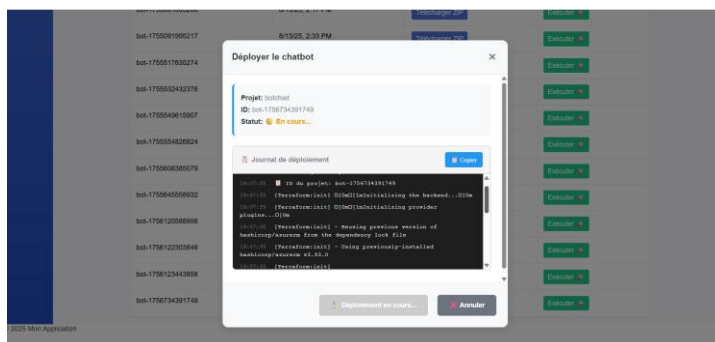


FIGURE 6.4 – le chatbot en cours de déploiement

6.6 Conclusion

Au terme de ce sprint, nous avons mis en place le processus complet de déploiement dynamique sur Azure, intégrant le formulaire de déploiement et l'exécution des fichiers Terraform. Le formulaire permet à l'utilisateur de fournir ses informations d'authentification Azure garantissant un déploiement personnalisé dans son propre abonnement.

Le processus de déploiement automatisé assure la création cohérente et répétable de toutes les ressources nécessaires, tout en gérant les erreurs et en informant l'administrateur et l'utilisateur du succès ou de l'échec.

Conclusion Générale

Au fil du temps, ce projet de fin d'études nous a menés à créer une plateforme entière, à la manière d'un Chatbot as a Service, dans le but d'automatiser, de simplifier la création, la paramétrisation et le déploiement de chatbots pour tous types d'environnement.

Chacun de ces sprints a été pensé pour aller plus loin, du cadrage, en passant par les formulaires dynamiques, jusqu'à la génération automatique des fichiers Terraform et leur déploiement sur Azure. Il nous a fallu structurer un formulaire dynamique, qui permettrait à l'utilisateur de renseigner et de nous remonter toutes les informations nécessaires, mais aussi les informations Azure (Client ID, Client Secret, Tenant ID, Subscription ID).

C'est en quelque sorte le "formulaire d'inscription" à ce type de déploiements : il permet à chacun de déployer son assistant sur son propre abonnement, tout en indiquant les informations essentielles pour que le script automatisé puisse fonctionner !

Nous avons également dû mettre en place un backend qui savait, à partir du formulaire, générer automatiquement les fichiers terraform, et qui sait ensuite orchestrer l'exécution de la commande de déploiement.

Ce système nous permet de créer de manière fiable et reproductible toutes les ressources qui seront nécessaires sur Azure, moteur de gestion des erreurs et des exceptions, et qui prévient ses utilisateurs et administrateurs pour un suivi clair, simple et efficace de l'état des déploiements réalisés.

De plus, la gestion des modèles et la personnalisation des chatbots ont contribué à évoluer l'expérience utilisateur en permettant de contrôler l'aspect et le fonctionnement du chatbot avant sa mise en service.

L'interface a pour objectif d'être intuitive et simple d'utilisation pour l'administrateur qui peut suivre et documenter l'ensemble du procédé afin de faciliter la maintenance et la reproduction des installations.

En résumé, ce projet constitue une solution intégrée et flexible offrant une contribution majeure en matière d'automatisation de rapidité et de fiabilité dans la création et le déploiement de chatbots. Il permet de réduire les erreurs humaines, d'assurer la cohérence des ressources, et de fournir aux utilisateurs et aux administrateurs un suivi clair et précis de l'ensemble des opérations.

Ainsi, ce projet illustre notre engagement envers l'automatisation, la flexibilité et l'amélioration continue, offrant une solution complète et moderne pour la création, la configuration et le déploiement de chatbots personnalisés dans un environnement cloud sécurisé.

Références bibliographiques

- [1] <https://www.jetbrains.com/idea/>
- [2] <https://code.visualstudio.com/>
- [3] <https://learn.microsoft.com/azure/app-service/>
- [4] <https://learn.microsoft.com/azure/azure-sql/>
- [5] <https://learn.microsoft.com/entra/>
- [6] <https://learn.microsoft.com/azure/ai-services/openai/>
- [7] <https://learn.microsoft.com/azure/devops/>
- [8] <https://angular.io/>
- [9] <https://spring.io/projects/spring-boot>
- [10] <https://jwt.io/>
- [11] <https://www.postman.com/>
- [12] <https://git-scm.com/>
- [13] <https://github.com/>



ECOLE SUPÉRIEURE PRIVÉE D'INGÉNIERIE ET DE TECHNOLOGIES

www.esprit.tn - E-mail : contact@esprit.tn

Siège Social : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889

Annexe : Z.I. Chotrana II - B.P. 160 - 2083 - Pôle Technologique - El Ghazala - Tél. : +216 70 685 685 - Fax. : +216 70 685 454